1. Blinking LEDs on 8051(Assembly)

ORG 00H ; Start of program memory  
  
MAIN:  
; Pattern 1: Turn ON all LEDs  
MOV P2, #0FFH  
ACALL DELAY  
MOV P2, #00H  
ACALL DELAY  
  
; Pattern 2: Alternate LEDs - 10101010  
MOV P2, #0AAH  
ACALL DELAY  
MOV P2, #00H  
ACALL DELAY  
  
; Pattern 3: Complementary alternate LEDs - 01010101  
MOV P2, #055H  
ACALL DELAY  
MOV P2, #00H  
ACALL DELAY  
  
SJMP MAIN  
  
; -------------------------  
; Delay Subroutine  
; --------------------------  
  
DELAY:  
MOV R2, #255  
LOOP1: MOV R3, #255  
LOOP2: DJNZ R3, LOOP2  
DJNZ R2, LOOP1  
RET  
  
END

2. Blinking LEDs on 8051 (EmbeddedC)

#include <REG51.H> // Include header for 8051 registers  
  
// Simple delay function  
void delay(int count) {  
    int i, j;  
    for(i = 0; i < count; i++) {  
        for(j = 0; j < 1000; j++); // Rough time delay  
    }  
}  
  
void main() {  
    while(1) {  
        // All LEDs ON  
        P2 = 0xFF;  
        delay(100);  
  
        // All LEDs OFF  
        P2 = 0x00;  
        delay(100);  
  
        // Alternate LEDs ON (10101010)  
        P2 = 0xAA;  
        delay(100);  
  
        // All LEDs OFF  
        P2 = 0x00;  
        delay(100);  
  
        // Complementary alternate LEDs ON (01010101)  
        P2 = 0x55;  
        delay(100);  
  
        // All LEDs OFF  
        P2 = 0x00;  
        delay(100);  
    }  
}

3. Display 00 to 99 on 7-segment with  
  
8051 (EmbeddedC)

#include <REG51.H> // 8051 register definitions  
  
#define SEGPORT P2       // Segment data port  
#define SS\_3 0x10         // Enable tens digit (e.g., P3.4 = 1)  
#define SS\_4 0x20         // Enable units digit (e.g., P3.5 = 1)  
  
// Common cathode 7-segment display segment codes for digits 0–9  
const unsigned char Lookuptable[] = {  
    0x3F, // 0  
    0x06, // 1  
    0x5B, // 2  
    0x4F, // 3  
    0x66, // 4  
    0x6D, // 5  
    0x7D, // 6  
    0x07, // 7  
    0x7F, // 8  
    0x6F  // 9  
};  
  
// Simple millisecond delay function  
void DELAY\_ms(unsigned int x) {  
    unsigned int i, j;  
    for (i = 0; i < x; i++)  
        for (j = 0; j < 500; j++); // Adjust for clock speed (rough estimate)  
}  
  
void main() {  
    int cnt, tens, units, i;  
  
    while (1) {  
        for (cnt = 0; cnt <= 99; cnt++) {  
            for (i = 0; i < 30; i++) { // Multiplexing delay loop  
  
                // Display tens digit  
                tens = cnt / 10;  
                P3 = SS\_3;                       // Enable tens digit (e.g., set P3.4 high)  
                SEGPORT = Lookuptable[tens];    // Send segment pattern  
                DELAY\_ms(1);                    // Brief display time  
                P3 = 0x00;                      // Disable both digits  
  
                // Display units digit  
                units = cnt % 10;  
                P3 = SS\_4;                       // Enable units digit (e.g., set P3.5 high)  
                SEGPORT = Lookuptable[units];   // Send segment pattern  
                DELAY\_ms(1);                    // Brief display time  
                P3 = 0x00;                      // Disable both digits  
            }  
        }  
    }  
}

4. Square Waveform with DAC 0808  
  
(EmbeddedC)

#include <reg51.h> // Include 8051 SFR (Special Function Register) definitions  
  
// Software delay in milliseconds  
void delay\_ms(unsigned long x) {  
    unsigned int i, j;  
    for (i = 0; i < x; i++) {  
        for (j = 0; j < 500; j++); // Adjust based on clock frequency  
    }  
}  
  
void main() {  
    while (1) {  
        P2 = 0x00;        // All LEDs OFF  
        delay\_ms(500);    // 500 ms delay  
  
        P2 = 0xFF;        // All LEDs ON  
        delay\_ms(500);    // 500 ms delay  
    }  
}

5. TriangularWaveform with DAC 0808  
  
(EmbeddedC)

#include <reg51.h>  
  
// Delay function: generates ~1ms delay per loop at ~11.0592 MHz clock  
void delay\_ms(unsigned int x) {  
    unsigned int i, j;  
    for (i = 0; i < x; i++)  
        for (j = 0; j < 1275; j++);  // Calibrated for ~1ms delay  
}  
  
void main() {  
    unsigned char i;  
  
    while (1) {  
        // Rising part of the triangular wave (0 to 255)  
        for (i = 0; i <= 255; i++) {  
            P2 = i;  
            delay\_ms(50);  
        }  
  
        // Falling part of the triangular wave (255 to 0)  
        for (i = 255; i > 0; i--) {  
            P2 = i;  
            delay\_ms(50);  
        }  
  
        P2 = 0;  // Reset to 0  
        delay\_ms(50);  
    }  
}

6. Display &#39;Nashik&#39; on 16x2 LCD  
  
(EmbeddedC)

#include <reg51.h>  
  
#define lcd P1            // LCD data bus connected to Port 1  
  
sbit rs = P2^2;           // Register Select pin  
sbit rw = P2^1;           // Read/Write pin  
sbit en = P2^0;           // Enable pin  
  
// Function prototypes  
void lcd\_init();  
void cmd(unsigned char);  
void dat(unsigned char);  
void lcd\_string(unsigned char \*);  
void delay(unsigned int);  
  
void main() {  
    lcd\_init();           // Initialize LCD  
    cmd(0x80);            // Set cursor to beginning of 1st line  
    lcd\_string("Nashik"); // Display string  
    while (1);            // Infinite loop  
}  
  
// LCD initialization  
void lcd\_init() {  
    cmd(0x38); // 2 lines, 5x7 matrix  
    cmd(0x0E); // Display ON, cursor ON  
    cmd(0x06); // Auto-increment cursor  
    cmd(0x01); // Clear display  
    delay(2);  
}  
  
// Send command to LCD  
void cmd(unsigned char x) {  
    lcd = x;  
    rs = 0;     // Command mode  
    rw = 0;     // Write mode  
    en = 1;  
    delay(2);  
    en = 0;  
}  
  
// Send data to LCD  
void dat(unsigned char y) {  
    lcd = y;  
    rs = 1;     // Data mode  
    rw = 0;     // Write mode  
    en = 1;  
    delay(2);  
    en = 0;  
}  
  
// Send string to LCD  
void lcd\_string(unsigned char \*s) {  
    while (\*s) {  
        dat(\*s++); // Send characters one by one  
    }  
}  
  
// Delay function (~1 ms per unit at 11.0592 MHz)  
void delay(unsigned int z) {  
    unsigned int i, j;  
    for (i = 0; i < z; i++)  
        for (j = 0; j < 1275; j++); // Roughly 1 ms  
}

7. Display &#39;KKWIEER&#39; on 16x2 LCD  
  
(EmbeddedC)

#include <reg51.h>  
  
#define lcd P1        // LCD data lines connected to Port 1  
sbit rs = P2^2;       // Register Select pin  
sbit rw = P2^1;       // Read/Write pin  
sbit en = P2^0;       // Enable pin  
  
void lcd\_init();  
void cmd(unsigned char);  
void dat(unsigned char);  
void lcd\_string(unsigned char \*);  
void delay(unsigned int);  
  
void main() {  
    lcd\_init();       // Initialize the LCD  
    cmd(0xC0);        // Move cursor to second line (0x80 + 0x40 = 0xC0)  
    lcd\_string("KKWIEER"); // Display text on second line  
    while (1);        // Infinite loop to keep displaying  
}  
  
void lcd\_init() {  
    cmd(0x38); // Function set: 2 lines, 5x7 matrix  
    cmd(0x0E); // Display ON, cursor ON  
    cmd(0x06); // Increment cursor automatically  
    cmd(0x01); // Clear display  
    delay(2);  
}  
  
void cmd(unsigned char x) {  
    lcd = x;  
    rs = 0;  // Command mode  
    rw = 0;  // Write mode  
    en = 1;  
    delay(2);  
    en = 0;  
}  
  
void dat(unsigned char y) {  
    lcd = y;  
    rs = 1;  // Data mode  
    rw = 0;  // Write mode  
    en = 1;  
    delay(2);  
    en = 0;  
}  
  
void lcd\_string(unsigned char \*s) {  
    while (\*s) {  
        dat(\*s++); // Send each character until null terminator  
    }  
}  
  
// Approximate delay (1 ms per unit at 11.0592 MHz crystal)  
void delay(unsigned int z) {  
    unsigned int i, j;  
    for (i = 0; i < z; i++)  
        for (j = 0; j < 1275; j++);  
}